

State complexity of multiple catenation

Pascal Caron, Jean-Gabriel Luque, and Bruno Patrou *

Département d'Informatique, Université de Rouen,
Avenue de l'Université,
76801 Saint-Étienne du Rouvray Cedex,
France

Abstract. We improve some results relative to the state complexity of the multiple catenation described by Gao and Yu. In particular we nearly divide by 2 the size of the alphabet needed for witnesses. We also give some refinements to the algebraic expression of the state complexity, which is especially complex with this operation. We obtain these results by using peculiar DFAs defined by Brzozowski.

1 Introduction

State complexity is a very active research area. It aims to determine the maximal size of a minimal automaton recognizing a language belonging to a given class. State complexity can be studied from the deterministic as well as non-deterministic point of view. Here, we only consider the deterministic case. Then, the state complexity of a regular language is the states number of its minimal DFA (Deterministic Finite Automaton). And the state complexity of a regular operation allows to compute the maximal size of any DFA obtained by applying this operation over regular languages, knowing their respective state complexities. Such operations can be elementary (see, as one of the first reference in this domain, [12]) or the result of some combinations (see, for example, [7], [3] or [11]). Sometimes, the computation of state complexities needs to use heavy tools of combinatorial, as in [2]. To have an expanded view of the domain, it is useful to refer to the surveys [6] and [5].

In [12], the authors are the first ones to study the state complexity of catenation. They prove $m2^n - 2^{n-1}$ to be the upper bound for the states number of a minimal DFA recognizing the catenation of two regular languages with respective state complexities m and n . And they propose a 3-letters witness reaching the bound. In [10], G. Jiraskova produces a 2-letters witness. In [8], the authors study a generalization by considering the sequential catenation of an arbitrary number α of regular languages. The upper bound they find is very intricate to write, its algebraic representation being growing with α . The witnesses they describe are defined over $(2\alpha - 1)$ -letters alphabets. In [1], J. Brzozowski shows that a particular family of DFAs can be used to produce witnesses in a very large number of cases.

In this paper, we focus on sequential catenation of α DFAs and our contributions are the following: first, we give a recursive definition of the state complexity which can be easily computed. Then, as our main result, we improve the set of witnesses by dramatically reducing the size of the alphabet from $2\alpha - 1$ to $\alpha + 1$. For this, we use DFAs issued from the Brzozowski family. Last, we conjecture it is possible to decrease the size of the alphabet until α (which should be optimal) still using Brzozowski DFAs. We test computationally our conjecture until 6 or 7 DFAs, and prove it when $\alpha = 2$ (giving here a positive issue to a remark made by Brzozowski who thought its family was deficient in this peculiar case) and $\alpha = 3$.

In section 2 are recalled the classical tools we need both in automata theory and in algebraic combinatorics. Section 3 is devoted to the presentation of the construction used for multiple catenation and to compute the upper bound for the state complexity of this construction. In section 4, we describe a family of α DFAs over an $(\alpha + 1)$ -letters alphabet and prove it to be a witness for the catenation of α regular languages. For the same operation, we give, in section 5, witnesses over α -letters alphabet when $\alpha = 2$ and $\alpha = 3$ and we conjecture these witnesses can be extended for any value of α .

2 Preliminaries

In all this paper, Σ denotes a finite alphabet. The set of all finite words over Σ is denoted by Σ^* . The empty word is denoted by ε . A language is a subset of Σ^* . The set of subsets of a finite set A is denoted by 2^A and $\#A$ denotes the cardinality of A . In the following, by abuse of notation, we often write q for any singleton $\{q\}$.

* {Pascal.Caron, Jean-Gabriel.Luque, Bruno.Patrou}@univ-rouen.fr

A finite automaton (FA) is a 5-tuple $A = (\Sigma, Q, I, F, \cdot)$ where Σ is the input alphabet, Q is a finite set of states, $I \subset Q$ is the set of initial states, $F \subset Q$ is the set of final states and \cdot is the transition function from $Q \times \Sigma$ to 2^Q . A FA is deterministic (DFA) if $\#I = 1$ and for all $q \in Q$, for all $a \in \Sigma$, $\#(q \cdot a) \leq 1$. Let a be a symbol of Σ . Let w be a word of Σ^* . The transition function is extended to any word by $q \cdot aw = \bigcup_{q' \in q \cdot a} q' \cdot w$ and $q \cdot \varepsilon = q$.

A symmetric use of the dot notation leads to the following definition. Let $w \cdot q = \{q' \mid q \in q' \cdot w\}$. We extend the dot notation to any set of states S by $S \cdot w = \bigcup_{s \in S} s \cdot w$ and $w \cdot S = \bigcup_{s \in S} w \cdot s$. A word $w \in \Sigma^*$ labels a successful path in a FA A if $I \cdot w \cap F \neq \emptyset$.

In this paper, we assume that all FA are complete which means that for all $q \in Q$, for all $a \in \Sigma$, $\#(q \cdot a) \geq 1$. A state q is accessible in a FA if there exists a word $w \in \Sigma^*$ such that $q \in I \cdot w$. The language recognized by a FA A is the set of words labeling a successful path in A . Two automata are said to be equivalent if they recognize the same language.

Let $D = (\Sigma, Q_D, i_D, F_D, \cdot)$ be a DFA. Two states q_1, q_2 of D are equivalent if for any word w of Σ^* , $q_1 \cdot w \in F_D$ if and only if $q_2 \cdot w \in F_D$. Such an equivalence is denoted by $q_1 \sim q_2$. A DFA is minimal if there does not exist any equivalent DFA with less states and it is well known that for any DFA, there exists a unique minimal equivalent one [9]. Such a minimal DFA can be obtained from D by computing the accessible part of the automaton $D/\sim = (\Sigma, Q_D/\sim, [i_D], F_D/\sim, \cdot)$ where for any $q \in Q_D$, $[q]$ is the \sim -class of the state q and for any $a \in \Sigma$, $[q] \cdot a = [q \cdot a]$. In a minimal DFA, any two distinct states are pairwise non-equivalent.

The states of a FA are often denoted with indexed symbols and arithmetic operations can be used to compute new index from given ones. Since this index allows to point to a state of the same FA, the operations are always done modulo the states number of the FA. This is recurrent in the paper and, in general, not explicitly mentioned.

The state complexity of a regular language L denoted by $\text{sc}(L)$ is the number of states of its minimal DFA. Let \mathcal{L}_n be the set of languages of state complexity n . The state complexity of a unary operation \otimes is the function sc_\otimes associating with an integer n the maximum of the state complexities of $(\otimes L)$ for $L \in \mathcal{L}_n$. A language $L \in \mathcal{L}_n$ is a witness (for \otimes) if $\text{sc}(\otimes L) = \text{sc}_\otimes(n)$. This can be generalized, and the state complexity of a k -ary operation \otimes is the k -ary function which associates with any tuple (n_1, \dots, n_k) the integer $\max\{\text{sc}(\otimes(L_1, \dots, L_k)) \mid L_i \in \mathcal{L}_{n_i}, \forall i \in [1, k]\}$. Then, a witness is a tuple $(L_1, \dots, L_k) \in (\mathcal{L}_{n_1} \times \dots \times \mathcal{L}_{n_k})$ such that $\text{sc}(\otimes(L_1, \dots, L_k)) = \text{sc}_\otimes(n_1, \dots, n_k)$. An important research area consists in finding witnesses for any $(n_1, \dots, n_k) \in \mathbb{N}^k$.

For example, let us consider the ternary operation \otimes defined for any three languages L_1, L_2, L_3 by $\otimes(L_1, L_2, L_3) = L_1 \cdot (L_2 \cdot L_3)$ and let h be its state complexity. Let f be the state complexity of \cdot . For any three integers n_1, n_2, n_3 , it holds $h(n_1, n_2, n_3) \leq f(f(n_1, n_2), n_3)$ [8]. In fact, applying the catenation on a witness does not produce a good candidate for a witness.

In [1], Brzozowski defines a family of languages that turns to be universal witnesses for several operations. The automata denoting these languages are called *Brzozowski automata*. We need some background to define these automata. We follow the terminology of [4]. Let $Q = \{0, \dots, n-1\}$ be a set. A *transformation* of the set Q is a mapping of Q into itself. If t is a transformation and i an element of Q , we denote by it the image of i under t . A transformation of Q can be represented by $t = [i_0, i_1, \dots, i_{n-1}]$ which means that $i_k = kt$ for each $0 \leq k \leq n-1$ and $i_k \in Q$. A *permutation* is a bijective transformation on Q . The *identity* permutation of Q is denoted by $\mathbb{1}$. A *cycle* of length $\ell \leq n$ is a permutation c , denoted by $(i_0, i_1, \dots, i_{\ell-1})$, on a subset $I = \{i_0, \dots, i_{\ell-1}\}$ of Q where $i_k c = i_{k+1}$ for $0 \leq k < \ell-1$ and $i_{\ell-1} c = i_0$. A *k-rotation* is obtained by composing k times the same cycle. In other word, we construct a k -rotation r_k from the cycle $(i_0, \dots, i_{\ell-1})$ by setting $i_j r_k = i_{j+k \bmod \ell}$ for $0 \leq j \leq \ell-1$. A *transposition* $t = (i, j)$ is a permutation on Q where $it = j$ and $jt = i$ and for every elements $k \in Q \setminus \{i, j\}$, $kt = k$. A *contraction* $t = \begin{pmatrix} i \\ j \end{pmatrix}$ is a transformation where $it = j$ and for every elements $k \in Q \setminus \{i\}$, $kt = k$. Then, a Brzozowski automaton is a complete DFA $(\Sigma, Q = \{0, \dots, n-1\}, 0, F = \{q_f = n-1\}, \cdot)$, where any letter of Σ induces one of the transformation among transposition, cycle over Q , contraction and identity. Let a, b, c, d be distinct symbols of Σ . As an example of Brzozowski automata (see Figure 1), let $W_n(a, b, c, d) = (\Sigma, Q_n, 0, \{q_f\}, \cdot)$ where $Q_n = \{0, 1, \dots, n-1\}$, the symbol a acts as the cycle $(0, 1, \dots, n-1)$, b acts as the transposition $(0, 1)$, c acts as the contraction $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and d acts as $\mathbb{1}$.

3 A bound for the state complexity of the multiple catenation

We first define a construction for the multiple catenation. We then compute an upper bound for the number of states of the resulting automaton.

Definition 1. (*Subset construction*) Let $A = (\Sigma, Q_A, I_A, F_A, \cdot_A)$ be an NFA. The "subset construction" consists to build the following DFA: $B = (\Sigma, Q, I, F, \cdot)$ where

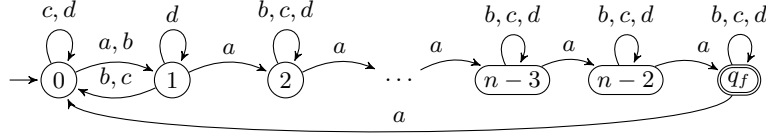


Fig. 1. The automaton $W_n(a, b, c, d)$

- $Q = 2^{Q_A}$
- $I = \{I_A\}$
- $F = \{X \in Q \mid X \cap F_A \neq \emptyset\}$
- $\forall (X, a) \in Q \times \Sigma, X \cdot a = \bigcup_{p \in X} (p \cdot_A a)$

Definition 2. Let $A = (\Sigma, Q_A, I_A, F_A, \cdot_A)$ and $B = (\Sigma, Q_B, I_B, F_B, \cdot_B)$ be two NFAs. We compute the NFA $A \cdot B = (\Sigma, Q, I, F, \cdot)$ as follows:

- $Q = Q_A \cup Q_B$
- $I = \begin{cases} I_A & \text{if } I_A \cap F_A = \emptyset \\ I_A \cup I_B & \text{otherwise} \end{cases}$
- $F = F_B$
- $p \cdot a = \begin{cases} p \cdot_B a & \text{if } p \in Q_B \\ p \cdot_A a & \text{if } p \cdot_A a \cap F_A = \emptyset \wedge p \in Q_A \\ p \cdot_A a \cup I_B & \text{otherwise} \end{cases}$

Lemma 1. $L(A \cdot B) = L(A) \cdot L(B)$

Proof. - Let $(u, v) \in L(A) \times L(B)$. If $u = \varepsilon$, $I_A \cap F_A \neq \emptyset$ and then $I_B \subset I$. We have $I_B \cdot_B v \cap F_B \neq \emptyset$, hence $I \cdot v \cap F \neq \emptyset$. We deduce $uv = v \in L(A \cdot B)$. If $u = u'a$ then there exists $p \in I_A \cdot_A u'$ such that $p \cdot_A a \cap F_A \neq \emptyset$. So we have $I_B \subset p \cdot a$. Moreover, $I_B \cdot_B v \cap F_B \neq \emptyset$. It follows $I_A \cdot u' \cdot a \cdot v \cap F \neq \emptyset$ and then $uv \in L(A \cdot B)$.
- Let $w \in L(A \cdot B)$. From the definition of I , we deduce that $I_B \subset I$ if and only if $\varepsilon \in L(A)$. If $I_B \subset I$ and $I_B \cdot w \cap F \neq \emptyset$ then, since the algorithm does not add any transition from automaton B to automaton A , we have $I_B \cdot_B w \cap F_B \neq \emptyset$. So, we have $w = uv$ with $u = \varepsilon \in L(A)$ and $v = w \in L(B)$.
If $I_B \cap I = \emptyset$, we have $I_A \cdot w \cap F \neq \emptyset$. From the definition of \cdot , we deduce that there exists $u', v \in \Sigma^*$, $a \in \Sigma$, $p \in Q_A$ such that $p \in I_A \cdot_A u'$ and $p \cdot_A a \cap F_A \neq \emptyset$ hence $u = u'a \in L(A)$. Furthermore, $I_B \subset p \cdot a$ and $I_B \cdot_B v \cap F_B \neq \emptyset$ imply $v \in L(B)$.

Let us consider a sequence of complete DFAs $A_1 = (\Sigma, Q_1, \{i_1\}, F_1, \cdot_1), \dots, A_\alpha = (\Sigma, Q_\alpha, \{i_\alpha\}, F_\alpha, \cdot_\alpha)$ we want to concatenate. We denote $A_{\bar{\alpha}} = (\Sigma, Q_{\bar{\alpha}}, I_{\bar{\alpha}}, F_{\bar{\alpha}}, \cdot_{\bar{\alpha}})$ the NFA defined by $(\dots((A_1 \cdot A_2) \cdot A_3) \dots) \cdot A_\alpha$. From previous lemma, we know that $L(A_1) \cdot L(A_2) \dots L(A_\alpha) = L(A_{\bar{\alpha}})$.

Each state of the DFA $A_{\bar{\alpha}} = (\Sigma, Q_{\bar{\alpha}}, I_{\bar{\alpha}}, F_{\bar{\alpha}}, \cdot_{\bar{\alpha}})$ obtained by applying the subset construction to the NFA $A_{\bar{\alpha}}$ can be partitioned and seen as a sequence of the form $(S_1, S_2, \dots, S_\alpha)$ where each S_j is a subset of Q_j . It is easy to see that if such a sequence corresponds to an accessible state of $A_{\bar{\alpha}}$, then it verifies the three following properties:

- P1. S_1 is a singleton (often assimilated to its unique element),
- P2. $\forall 0 < k < \alpha, (S_k = \emptyset \Rightarrow S_{k+1} = \emptyset)$ (to access a DFA, we must go through its predecessors),
- P3. $\forall 0 < k < \alpha, (S_k \cap F_k \neq \emptyset \Rightarrow i_{k+1} \in S_{k+1})$ (because of the transitions built in Definition 2).

A sequence which verifies properties P1, P2 and P3 is called a *valid* sequence. A state associated to such a sequence is called a *valid* state. We now evaluate an upper bound for the number of valid states, that is an upper bound for the state complexity of multiple catenation.

3.1 Counting states

Let us notice that the maximum number of valid states is reached when each DFA has only one final state. So the number of valid states is upper bounded by $\#(\mathcal{T}_\alpha)$ where \mathcal{T}_α is the set of valid sequences (S_1, \dots, S_α) denoting the states of the catenation of α such automata of size n_1, \dots, n_α .

For any $j \in \mathbb{N} \setminus \{0\}$, let \mathcal{T}_j^+ (resp \mathcal{T}_j^-) be the subset of \mathcal{T}_j constituted with the sequences of non empty sets (S_1, \dots, S_j) with $q_f \in S_j$ (resp. $q_f \notin S_j$). A fast examination of the elements of \mathcal{T}_α gives

Lemma 2. *The set \mathcal{T}_α is the disjoint union*

$$\mathcal{T}_\alpha = \mathcal{T}_\alpha^+ \uplus \biguplus_{j=1}^{\alpha} \{(S_1, \dots, S_j, \emptyset, \dots, \emptyset) : (S_1, \dots, S_j) \in \mathcal{T}_j^-\}. \quad (1)$$

Notice that for each $1 < j \leq \alpha$, \mathcal{T}_j^- splits into two disjoint sets

$$\mathcal{T}_j^- = \mathcal{T}_j^{--} \uplus \mathcal{T}_j^{+-} \quad (2)$$

where

$$\mathcal{T}_j^{--} = \{(S_1, \dots, S_j) \in \mathcal{T}_j^- \mid (S_1, \dots, S_{j-1}) \in \mathcal{T}_{j-1}^-\}, \quad (3)$$

and

$$\mathcal{T}_j^{+-} = \{(S_1, \dots, S_j) \in \mathcal{T}_j^- \mid (S_1, \dots, S_{j-1}) \in \mathcal{T}_{j-1}^+\}. \quad (4)$$

Also \mathcal{T}_j^+ splits into two disjoint sets

$$\mathcal{T}_j^+ = \mathcal{T}_j^{-+} \uplus \mathcal{T}_j^{++} \quad (5)$$

where

$$\mathcal{T}_j^{-+} = \{(S_1, \dots, S_j) \in \mathcal{T}_j^+ \mid (S_1, \dots, S_{j-1}) \in \mathcal{T}_{j-1}^-\}, \quad (6)$$

and

$$\mathcal{T}_j^{++} = \{(S_1, \dots, S_j) \in \mathcal{T}_j^+ \mid (S_1, \dots, S_{j-1}) \in \mathcal{T}_{j-1}^+\}. \quad (7)$$

Proposition 1. *We have the following crossing recurrence*

$$\begin{cases} \#\mathcal{T}_1^- = n_1 - 1, \\ \#\mathcal{T}_1^+ = 1, \\ \#\mathcal{T}_j^- = (2^{n_j-1} - 1)(\#\mathcal{T}_{j-1}^-) + 2^{n_j-2}(\#\mathcal{T}_{j-1}^+) \\ \#\mathcal{T}_j^+ = 2^{n_j-1}(\#\mathcal{T}_{j-1}^-) + 2^{n_j-2}(\#\mathcal{T}_{j-1}^+) = \#\mathcal{T}_j^- + \#\mathcal{T}_{j-1}^-. \end{cases} \quad (8)$$

Furthermore, we have

$$\#\mathcal{T}_\alpha = \sum_{j=1}^{\alpha} \#\mathcal{T}_j^- + \#\mathcal{T}_j^+. \quad (9)$$

Proof. It suffices to remark that formulas (3), (4), (6), (7) imply

$$\#\mathcal{T}_j^{--} = (2^{n_j-1} - 1)(\#\mathcal{T}_{j-1}^-), \quad \#\mathcal{T}_j^{+-} = 2^{n_j-2}(\#\mathcal{T}_{j-1}^+), \quad (10)$$

$$\#\mathcal{T}_j^{-+} = 2^{n_j-1}(\#\mathcal{T}_{j-1}^-), \quad \#\mathcal{T}_j^{++} = 2^{n_j-2}(\#\mathcal{T}_{j-1}^+). \quad (11)$$

Formula (9) comes immediately from Lemma 2.

Example 1. Applying proposition 1, we find after simplification

$$\#\mathcal{T}_2 = 2^{n_2-1} (2n_1 - 1)$$

$$\#\mathcal{T}_3 = n_1 - 1 + \frac{3}{8} 2^{n_3+n_2} (2n_1 - 1) + 2^{n_3} (1 - n_1).$$

$$\begin{aligned} \#\mathcal{T}_4 = & 2^{n_4} n_1 + \frac{9}{16} 2^{n_4+n_3+n_2} n_1 - \frac{3}{4} 2^{n_4+n_3} n_1 + 2^{n_2-1} n_1 - \frac{1}{4} 2^{n_2} - 2^{n_4} - \frac{9}{32} 2^{n_4+n_3+n_2} + \frac{1}{4} 2^{n_4+n_1} \\ & + \frac{3}{4} 2^{n_4+n_3} - 2^{n_4-1+n_2} n_1. \end{aligned}$$

$$\begin{aligned} \#\mathcal{T}_5 = & -1 + n_1 - \frac{3}{16} 2^{n_3+n_2} + \frac{3}{8} 2^{n_3+n_2} n_1 + 2^{n_3-1} - 2^{n_5-1+n_3} + \frac{3}{16} 2^{n_5+n_4+n_2} - \frac{27}{128} 2^{n_5+n_4+n_3+n_2} \\ & + \frac{9}{16} 2^{n_5+n_4+n_3} + 2^{n_5} - \frac{3}{4} 2^{n_5+n_4} + \frac{3}{16} 2^{n_5+n_3+n_2} - \frac{1}{4} 2^{n_5+n_2} - 2^{n_3-1} n_1 + 2^{n_5-1+n_3} n_1 \\ & - \frac{3}{8} 2^{n_5+n_4+n_2} n_1 + \frac{27}{64} 2^{n_5+n_4+n_3+n_2} n_1 - \frac{9}{16} 2^{n_5+n_4+n_3} n_1 - 2^{n_5} n_1 - \frac{3}{8} 2^{n_5+n_3+n_2} n_1 + \frac{3}{4} 2^{n_5+n_4} n_1 \end{aligned}$$

3.2 Expanded formula

As one can see, the first values of $\#\mathcal{T}_n$ can be recursively computed but seem to be tedious. Some regularities can be observed which allow us to propose a combinatorial description of $\#\mathcal{T}_n$.

For the sake of simplicity we consider the formal variables x_j for $j > 0$, y and z and define the multivariable polynomials s_j^- , s_j^+ and s_j for $j \geq 0$.

$$\begin{aligned} s_0^- &= z, s_0^+ = 2y, \\ s_j^- &= (x_j - 1)s_{j-1}^- + \frac{1}{2}x_j s_{j-1}^+, \quad \text{for } j > 0 \text{ and} \\ s_j^+ &= x_j s_{j-1}^- + \frac{1}{2}x_j s_{j-1}^+ = s_j^- + s_{j-1}^-, \quad \text{for } j > 0. \end{aligned}$$

We set also

$$s_{\alpha-1} = s_{\alpha-1}^+ + \sum_{j=0}^{\alpha-1} s_j^-. \quad (12)$$

Notice that we recover $\#\mathcal{T}_\alpha$ from $s_{\alpha-1}$ by setting $x_j = 2^{n_{j+1}-1}$ for $1 \leq j \leq \alpha - 1$, $y = \frac{1}{2}$, and $z = n_1 - 1$. For technical reasons, we will use the polynomial

$$r_{\alpha-1} = s_{\alpha-1}|_{x_{\alpha-1} \rightarrow \frac{1}{2}x_{\alpha-1}}. \quad (13)$$

where $P|_{x \rightarrow t}$ means that each occurrence of the variable x is replaced by the expression t in the polynomial P .

Example 2.

1. $r_0 = 2y + z$
2. $r_1 = x_1 y + x_1 z$
3. $r_2 = z + \frac{3}{2}x_2 x_1 y + \frac{3}{2}x_2 x_1 z - x_2 z$
4. $r_3 = \frac{9}{4}x_3 x_2 x_1 y + \frac{9}{4}x_3 x_2 x_1 z - x_3 x_1 y + x_1 y + x_1 z - \frac{3}{2}x_3 x_2 z + x_3 z - x_3 x_1 z$
5. $r_4 = -\frac{9}{4}x_4 x_3 x_2 z - \frac{3}{2}x_4 x_3 x_1 y - \frac{3}{2}x_4 x_3 x_1 z - \frac{3}{2}x_4 x_2 x_1 y - \frac{3}{2}x_4 x_2 x_1 z + z + \frac{27}{8}x_4 x_3 x_2 x_1 y + \frac{27}{8}x_4 x_3 x_2 x_1 z + \frac{3}{2}x_4 x_3 z + x_4 x_2 z + x_4 x_1 y + x_4 x_1 z + \frac{3}{2}x_2 x_1 z + \frac{3}{2}x_2 x_1 y - x_2 z - x_4 z$

It is easy to check that

$$r_{\alpha-1} = \sum_{j=0}^{\alpha-1} s_j^-. \quad (14)$$

Let us recall some notation. A composition is a finite list of positive integers $c = (c_1, \dots, c_\ell)$. When there is no ambiguity, we denote c by $c_1 \cdots c_\ell$. The length of the composition is $|c| = \ell$. We denote by $c \models n$ if and only if $c_1 + \cdots + c_\ell = n$. We define also $\Theta(c) = \#\{j \mid c_j = 1, 1 \leq j \leq \ell - 1\}$, $[c] = x_1 x_{1+c_1} x_{1+c_1+c_2} \cdots x_{1+c_1+\cdots+c_{\ell-1}}$ for $\ell > 0$ and $[()] = 0$.

Example 3. Consider $c = 212113$, one has $c \models 10$, $|c| = 6$, $\Theta(c) = 3$, $[c] = x_1 x_3 x_4 x_6 x_7 x_8$.

We define the polynomials

$$m_i = \sum_{c \models i} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c]. \quad (15)$$

Lemma 3. For $i > 1$ we have

$$\begin{cases} m_0 = 0 \\ m_1 = x_1 \\ m_i = \left(\frac{3}{2}x_i - 1\right) m_{i-1} + \frac{1}{2}x_i m_{i-2}. \end{cases}$$

Proof. Let $c \models i$. If $c = c'1$ then $c' \models i - 1$, $|c'| = |c| - 1$, $[c] = [c']x_i$ and

$$\Theta(c) = \begin{cases} \Theta(c') + 1 & \text{if } c_{|c'|} = 1 \\ \Theta(c') & \text{if } c_{|c'|} \neq 1 \end{cases} \quad (16)$$

If $c = c'p$ with $p > 1$ then setting $c'' = c'(p-1)$, we have $c'' \models i-1$ with $|c''| = |c|$, $[c] = [c'']$ and $\Theta(c'') = \Theta(c)$. According to the previous remarks, the sum splits as follows

$$\begin{aligned}
\mathbf{m}_i &= \sum_{\substack{c \models i \\ c|c| = c|c|-1=1}} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + \sum_{\substack{c \models i \\ c|c|=1, c|c|-1 > 1}} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + \sum_{\substack{c \models i \\ c|c| \neq 1}} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] \\
&= \sum_{\substack{c \models i-1 \\ c|c|=1}} (-1)^{|c|+1+i} \left(\frac{3}{2}\right)^{\Theta(c)+1} [c]x_i + \sum_{\substack{c \models i-1 \\ c|c| > 1}} (-1)^{|c|+1+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c]x_i + \sum_{c \models i-1} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] \\
&= \left(\frac{3}{2}x_i - 1\right) \sum_{\substack{c \models i-1 \\ c|c|=1}} (-1)^{|c|+i+1} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + (x_i - 1) \sum_{\substack{c \models i-1 \\ c|c| > 1}} (-1)^{|c|+i+1} \left(\frac{3}{2}\right)^{\Theta(c)} [c] \\
&= \left(\frac{3}{2}x_i - 1\right) \sum_{c \models i-1} (-1)^{|c|+i+1} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + \frac{1}{2}x_i \sum_{\substack{c \models i-1 \\ c|c| > 1}} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c]
\end{aligned} \tag{17}$$

And similarly, we also have

$$\sum_{\substack{c \models i-1 \\ c|c| > 1}} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] = \sum_{c \models i-2} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c]. \tag{18}$$

We deduce

$$\mathbf{m}_i = \left(\frac{3}{2}x_i - 1\right)\mathbf{m}_{i-1} + \frac{1}{2}x_i\mathbf{m}_{i-2}. \tag{19}$$

Lemma 4. Let \mathbf{s}_i^y be the coefficient of y in \mathbf{s}_i^- . We have

$$\mathbf{s}_i^y = \sum_{c \models i} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c]$$

Proof. It suffices to remark that \mathbf{s}_i^y satisfies the same recurrence and initial conditions as \mathbf{m}_i .

The following result is obtained in a very similar way.

Lemma 5. Let \mathbf{s}_i^z be the coefficient of z in \mathbf{s}_i^- . We have

$$\mathbf{s}_i^z = \sum_{c \models i+1} (-1)^{|c|+i+1} \left(\frac{3}{2}\right)^{\tilde{\Theta}(c)} \{c\}$$

where $\{c\} = x_{c_1}x_{c_1+c_2} \cdots x_{c_1+\dots+c_{|c|-1}}$ for any non-empty list c and $\tilde{\Theta}(c) = \#\{i \in \{2, \dots, |c|-1\} \mid c_i = 1\}$.

Proof. The proof follows the same pattern as in Lemma 3 and 4, since the sequence of the \mathbf{s}_i^z admits the same recurrence as the sequence of the \mathbf{s}_i^y but with the initial conditions $\mathbf{s}_0^z = 1$ and $\mathbf{s}_1^z = x_1 - 1$.

From Lemmas 4 and 5, one obtains

$$\mathbf{s}_i^- = y \sum_{c \models i} (-1)^{|c|+i} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + z \sum_{c \models i+1} (-1)^{|c|+i+1} \left(\frac{3}{2}\right)^{\tilde{\Theta}(c)} \{c\}. \tag{20}$$

Example 4. The compositions of 4 and 5 are summarized in the following tables:

c	$\Theta(x)$	$[c]$	c	$\tilde{\Theta}(x)$	$\{c\}$
4	0	x_1	5	0	1
31	0	x_1x_4	41	0	x_4
13	1	x_1x_2	14	0	x_1
211	1	$x_1x_3x_4$	32	0	x_3
121	1	$x_1x_2x_4$	23	0	x_2
112	2	$x_1x_2x_3$	311	1	x_3x_4
22	0	x_1x_3	131	0	x_1x_4
1111	3	$x_1x_2x_3x_4$	113	1	x_1x_2
			221	0	x_2x_4
			212	1	x_2x_3
			122	0	x_1x_3
			2111	2	$x_2x_3x_4$
			1211	1	$x_1x_3x_4$
			1121	1	$x_1x_2x_4$
			1112	2	$x_1x_2x_3$
			11111	3	$x_1x_2x_3x_4$

So we have

$$\begin{aligned} \mathbf{s}_4^- = & y \left(-x_1 + x_1x_4 + \frac{3}{2}x_1x_2 - \frac{3}{2}x_1x_3x_4 - \frac{3}{2}x_1x_2x_4 - \frac{9}{4}x_1x_2x_3 + \frac{27}{8}x_1x_2x_3x_4 \right) + \\ & z \left(1 - x_4 - x_1 - x_2 - x_3 + \frac{3}{2}x_3x_4 + x_1x_4 + \frac{3}{2}x_1x_2 \right. \\ & \left. + x_2x_4 + \frac{3}{2}x_2x_3 + x_1x_3 - \frac{9}{4}x_2x_3x_4 - \frac{3}{2}x_1x_3x_4 - \frac{3}{2}x_1x_2x_4 - \frac{9}{4}x_1x_2x_3 + \frac{27}{8}x_1x_2x_3x_4 \right). \end{aligned}$$

Theorem 1. *The number of valid states in the catenation of α automata of size $n_1, n_2, \dots, n_\alpha$ is bounded by the number obtained by setting $x_1 = 2^{n_2-1}, \dots, x_{\alpha-2} = 2^{n_{\alpha-1}-1}, x_{\alpha-1} = 2^{n_\alpha}, y = \frac{1}{2}$ and $z = n_1 - 1$ in the polynomial*

$$\mathbf{r}_{\alpha-1} = y \sum_{\substack{c=c'm \models \alpha-1 \\ m \text{ odd}}} (-1)^{|c|+\alpha-1} \left(\frac{3}{2}\right)^{\Theta(c)} [c] + z \sum_{\substack{c=c'm \models k \\ m \text{ odd}}} (-1)^{|c|+\alpha} \left(\frac{3}{2}\right)^{\tilde{\Theta}(c)} \{c\}. \quad (21)$$

Proof. From (12) and (13) one has only to prove formula (21). Suppose that $c \models p$, we have $[c1] = [c2] = \dots = [c(\alpha-p-1)]$ and $\Theta(c1) = \dots = \Theta(c(\alpha-p-1))$. Hence, by telescoping, the coefficient of the monomial $[c(\alpha-p-1)]$ in $\sum_{i=0}^{\alpha-1} \mathbf{s}_i^y$ equals 0 if $\alpha-p-1$ is even and equals $(-1)^{|c|+\alpha-1} \left(\frac{3}{2}\right)^{\Theta(c(\alpha-1))}$ if $\alpha-p-1$ is odd. Hence,

$$\sum_{i=0}^{\alpha-1} \mathbf{s}_i^y = \sum_{\substack{c=c'm \models \alpha-1 \\ m \text{ odd}}} (-1)^{|c|+\alpha-1} \left(\frac{3}{2}\right)^{\Theta(c)} [c]. \quad (22)$$

For similar reasons, we find

$$\sum_{i=0}^{\alpha-1} \mathbf{s}_i^z = \sum_{\substack{c=c'm \models \alpha \\ m \text{ odd}}} (-1)^{|c|+\alpha} \left(\frac{3}{2}\right)^{\tilde{\Theta}(c)} \{c\}. \quad (23)$$

Since

$$\sum_{i=0}^{\alpha-1} \mathbf{s}_i^- = y \sum_{i=0}^{\alpha-1} \mathbf{s}_i^y + z \sum_{i=0}^{\alpha-1} \mathbf{s}_i^z, \quad (24)$$

Equation 14 allows to conclude.

Example 5. The following tables summarize the compositions of 3 and 4 such that the last entry is odd:

c	$\Theta(x)$	$[c]$	c	$\tilde{\Theta}(x)$	$\{c\}$
3	0	x_1	31	0	x_3
21	1	x_1x_3	13	0	x_1
111	2	$x_1x_2x_3$	211	1	x_2x_3
			121	0	x_1x_3
			1111	2	$x_1x_2x_3$

So we obtain

$$r_3 = y(x_1 + \frac{9}{4}x_1x_2x_3 - x_1x_3) + (x_1 + \frac{9}{4}x_1x_2x_3 - \frac{3}{2}x_2x_3 - x_1x_3 + x_3)z$$

and then the cardinal of \mathcal{T}_4 is

$$\frac{1}{2}(2^{n_2-1} + \frac{9}{4}2^{n_2+n_3+n_4-2} - 2^{n_2+n_4-1}) + (2^{n_2-1} + \frac{9}{4}2^{n_2+n_3+n_4-2} - \frac{3}{2}2^{n_3+n_4-1} - 2^{n_2+n_4-1} + 2^{n_4})(n_1 - 1).$$

4 A $(\alpha + 1)$ -letters witness for the catenation of α automata

In this section, we give a family of witnesses automata for the multiple catenation. These witnesses are Brzozowski automata computed with the operations given in Table 1. Let us recall that for an automaton, $\mathbb{1}$ stands for the identity, p for the cycle $(0, \dots, n-1)$ on $Q = \{0, \dots, n-1\}$, t for the transposition $(0, 1)$ of the two first states and c for the contraction of state 1 to state 0.

	A_1	A_2	A_3	\dots	A_α
σ_1	t	c	$\mathbb{1}$		$\mathbb{1}$
σ_2	p	t	c		
σ_3	$\mathbb{1}$	p	t	c	$\mathbb{1}$
\dots					
σ_α					c
$\sigma_{\alpha+1}$	$\mathbb{1}$				t
					p

Table 1. Brzozowski witnesses for the multiple catenation

As a consequence, for any automaton A_k , σ_{k-1} acts as the contraction $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, σ_k acts as the transposition $(0, 1)$ and σ_{k+1} acts as the cycle $(0, \dots, n_k - 1)$ (see figure 2). Notice that there is no contraction for the automaton A_1 .

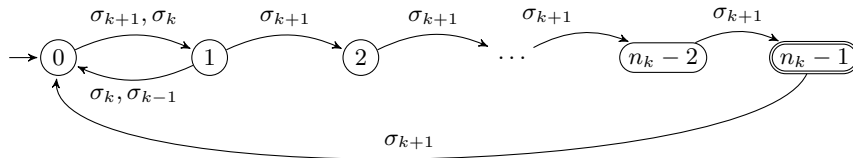


Fig. 2. The DFA A_k without the identity transitions

Let us recall that each state is associated to a sequence. We want to prove that, for our family of automata, the size of the minimal DFA for the multiple catenation is the number of valid sequences.

Proposition 2. *Two distinct states $s = (S_1, \dots, S_\alpha)$ and $s' = (S'_1, \dots, S'_\alpha)$ are not equivalent.*

Proof. By induction on $\mu = \max(\{k \in [1, \alpha] | S_k \neq S'_k\})$. If $k = \alpha$ then there exists j such that $q_j \in S_\alpha \oplus S'_\alpha$ and the word $(\sigma_{\alpha+1})^{n_\alpha-j}$ separates s and s' . If $k < \alpha$ then there exists j such that $q_j \in S_k \oplus S'_k$. The word $(\sigma_{k+1})^{n_k-j} \sigma_k (\sigma_{k+2})^{n_{k+1}-1}$ sends s and s' , respectively, to two states $t = (T_1, \dots, T_\alpha)$ and $t' = (T'_1, \dots, T'_\alpha)$ such that $0 \in T_{k+1} \oplus T'_{k+1}$. The states t and t' being non equivalent by the induction hypothesis, s and s' also are.

We now investigate the accessibility problem. The main difficulty appears when we have to access some valid state s by using a contraction. In such a situation, it is a bit technical to find the predecessor of s . Indeed, a

contraction on a DFA implies a transposition and a permutation on the two previous DFAs with some possible disturbances due to the property $P3$ (when a final state is reached in some DFA, the initial state of the next one is also reached). To solve this difficulty, we need some technical lemmas. Let A_k be an automaton defined in table 1. For any state q of A_k , $p \cdot q$ stands for $\sigma_{k+1} \cdot q$ and $t \cdot q$ stands for $\sigma_k \cdot q$. As usual, for any set of states S , we denote $p \cdot S = \bigcup_{q \in S} p \cdot q$ and $t \cdot S = \bigcup_{q \in S} t \cdot q$. These notations allows to shorten some expressions by omitting unambiguous indexes.

Definition 3. For any state $s = (i, S_2, \dots, S_{k-1}, S_k, S_{k+1}, \dots, S_\alpha)$, we define the two following transformations:

$$\begin{aligned} - \tau_k \cdot s &= (i, S_2, \dots, S_{k-3}, p \cdot S_{k-2}, t \cdot S_{k-1}, S_k \cup \{1\}, S_{k+1}, \dots, S_\alpha), \forall k \in [2, \alpha] \\ - \nu_k \cdot s &= (i, S_2, \dots, p^{\min(S_{k-3})+1} \cdot S_{k-3}, p \cdot (S_{k-2} \setminus \{0\}), t \cdot S_{k-1}, S_k \cup \{1\}, S_{k+1}, \dots, S_\alpha), \forall k \in [4, \alpha] \\ \nu_3 \cdot s &= \begin{cases} (p \cdot i, t \cdot S_2, S_3 \cup \{1\}, S_4, \dots, S_\alpha) & \text{if } i > 0 \\ (0, t \cdot S_2, S_3 \cup \{1\}, S_4, \dots, S_\alpha) & \text{if } i = 0 \end{cases} \end{aligned}$$

Lemma 6. Any valid state s satisfies the following properties:

1. $\forall k \in [2, \alpha]$, $\tau_k \cdot s$ is a valid state if and only if $(0 \notin S_{k-2} \text{ or } 1 \in S_{k-1})$ and $(q_f \notin S_{k-3} \text{ or } 1 \in S_{k-2})$. (P_τ)
2. $\forall k \in [4, \alpha]$, $\nu_k \cdot s$ is a valid state if and only if $1 \in S_{k-2}$ and $(q_f \notin S_{k-4} \text{ or } 1 \in S_{k-3})$. (P_ν)
3. $\nu_3 \cdot s$ is always a valid state.

Proof. 1. Property $P1$ is clear from the definition of τ . Property $P2$ is deduced from the fact that s is a valid state and that the size of any set of s can not be decreased by τ_k . Property $P3$ comes, for $k-1$, from the first parenthesis which asserts that it is not possible to have simultaneously $q_f \in p \cdot S_{k-2}$ ($0 \in S_{k-2}$) and $0 \notin e \cdot S_{k-1}$ ($1 \notin S_{k-1}$), for $k-2$, from the second parenthesis which asserts that it is not possible to have simultaneously $q_f \in S_{k-3}$ and $0 \notin p \cdot S_{k-2}$ ($1 \notin S_{k-2}$), and for all the other sets, from the fact that s is a valid state.

2. Property $P1$ is clear from the definition of ν . Property $P2$ is deduced from the fact that s is a valid state and because $1 \in S_{k-2}$ involves $S_{k-2} \neq \emptyset$. Property $P3$ comes, for $k-3$, from the last parenthesis which asserts that it is not possible to have simultaneously $q_f \in S_{k-4}$ (which implies $j_0 = 0$) and $0 \notin p^{j_0+1} \cdot S_{k-3}$ ($1 \notin S_{k-3}$), for $k-2$, from the fact that $0 \in p \cdot (S_{k-2} \setminus \{0\})$ ($1 \in S_{k-2}$), for $k-1$, from the fact that $q_f \notin p \cdot (S_{k-2} \setminus \{0\})$ and, for all the other sets, from the fact that s is a valid state.
3. clear from the definition.

Remark 1. The previous proof of properties P_τ and P_ν can be admitted for the first values of k , if we accept the convention that each time a set S_j is considered with $j < 1$, this set is assimilated to \emptyset . With this convention, the properties can be simplified as:

- $\tau_3 \cdot s$ is a valid state if and only if $(i \neq 0 \text{ or } 1 \in S_2)$.
- $\tau_2 \cdot s$ is always a valid state.
- $\nu_4 \cdot s$ is a valid state if and only if $1 \in S_2$.

This convention is often implicitly used in the following.

A composition of such transformations is denoted by a word over the alphabet $\Pi = \{\tau_k\}_{k \in [2, \alpha]} \cup \{\nu_k\}_{k \in [3, \alpha]}$.

Lemma 7. For any valid state $s = (i, S_2, \dots, S_{k-1}, S_k, S_{k+1}, \dots, S_\alpha)$ such that $0 \in S_k$ et $1 \notin S_k$ we have:

1. if $\tau_k \cdot s$ is a valid state then $(\tau_k \cdot s) \cdot \sigma_{k-1} = s$.
2. if $\nu_k \cdot s$ is a valid state and $(0 \in S_{k-2} \text{ and } 1 \notin S_{k-1})$ then $(\nu_k \cdot s) \cdot \sigma_{k-1} (\sigma_{k-2})^{\min(S_{k-3})+1} = s$ (for $k > 3$).
3. if $i > 0$ then $(\nu_3 \cdot s) \cdot \sigma_2 = s$.
4. if $i = 0$ and $1 \notin S_2$ then $(\nu_3 \cdot s) \cdot \sigma_2 \sigma_1 = s$.

Proof. 1. It is clear from the definition of τ_k that its action is cancelled by a transition labelled σ_{k-1} .
2. Let us first notice, as for τ_k , that σ_{k-1} cancels the action of ν_k on the sets S_{k-2} , S_{k-1} and S_k . Then, $(\sigma_{k-2})^{\min(S_{k-3})+1}$ cancels the action of ν_k on S_{k-3} and does not modify S_{k-2} and S_{k-1} if $1 \in S_{k-2}$ (which is true according to (P_ν)) and $1 \notin S_{k-1}$ (which is true by hypothesis).
3. If $i > 0$, σ_2 cancels the action of ν_3 .
4. If $i = 0$, σ_2 cancels the action of ν_3 on S_3 and S_2 but sends i in 1. The action of σ_1 allows us to send i in 0 without modifying S_3 and S_2 (because $1 \notin S_2$).

In the following lemma, let us write $\Sigma_k = \{\sigma_1, \dots, \sigma_k\}$.

Lemma 8. *When $k \geq 2$, for any valid state $s = (i, S_2, \dots, S_{k-1}, S_k, S_{k+1}, \dots, S_\alpha)$ such that $0 \in S_k$ and $1 \notin S_k$ there exists a couple $(u, v) \in (\Pi^*, (\Sigma_{k-2})^*)$ such that $s' = u \cdot s$ is a valid state of the form $(i', S'_2, \dots, S'_{k-1}, S_k \cup \{1\}, S_{k+1}, \dots, S_\alpha)$ and $s' \cdot \sigma_{k-1}v = s$.*

Proof. By induction on k . Multiple cases can appear. Someones, labeled (B), are bases cases. Other ones, labeled (IH), use the induction hypothesis.

- If $1 \in S_{k-1}$ we distinguish two cases:
 - (B1) If $q_f \notin S_{k-3}$ or $1 \in S_{k-2}$, the desired couple (u, v) is (τ_k, ε) . Indeed, $\tau_k \cdot s$ is a valid state by (P_τ) , of the announced form by definition of τ_k and $(\tau_k \cdot s) \cdot \sigma_{k-1} = s$ by lemma 7.
 - (IH1) If $q_f \in S_{k-3}$ (which implies $0 \in S_{k-2}$) and $1 \notin S_{k-2}$ then, by induction hypothesis, there exists a couple $(u', v') \in (\Pi^*, (\Sigma_{k-4})^*)$ such that $s'' = u' \cdot s$ is a valid state of the form $(i'', S''_2, \dots, S''_{k-3}, S_{k-2} \cup \{1\}, S_{k-1}, \dots, S_\alpha)$ and $s'' \cdot \sigma_{k-3}v' = s$. Since $S''_{k-1} = S_{k-1}$ and $1 \in S''_{k-2}$, we are taken back to point (B1) and the desired couple (u, v) is obtained by composing (τ_k, ε) with (u', v') , which gives $(\tau_k u', \sigma_{k-3}v')$.
- (B2) If $1 \notin S_{k-1}$ and $0 \notin S_{k-2}$ (and so $q_f \notin S_{k-3}$) then we follow a similar reasoning to the one used for case (B1) to find $(u, v) = (\tau_k, \varepsilon)$.
- If $1 \notin S_{k-1}$ and $0, 1 \in S_{k-2}$ we distinguish two cases:
 - (B3) If $q_f \notin S_{k-4}$ or $1 \in S_{k-3}$ then the desired couple (u, v) is $(\nu_k, (\sigma_{k-2})^{j_0+1})$, where

$$j_0 = \begin{cases} \min(S_{k-3}) & \text{if } k > 3 \\ -1 & \text{if } k = 3 \text{ and } i > 0 \\ 0 & \text{if } k = 3 \text{ and } i = 0 \end{cases}$$

Indeed, $\nu_k \cdot s$ is a state by (P_ν) , of the announced form by the definition of ν_k and $(\nu_k \cdot s) \cdot \sigma_{k-1}(\sigma_{k-2})^{j_0+1} = s$ by lemma 7.

- (IH2) If $q_f \in S_{k-4}$ (which implies $0 \in S_{k-3}$) and $1 \notin S_{k-3}$ then, by induction hypothesis, there exists a couple $(u', v') \in (\Pi^*, (\Sigma_{k-5})^*)$ such that $s'' = u' \cdot s$ is a valid state of the form $(i'', S''_2, \dots, S''_{k-4}, S_{k-3} \cup \{1\}, S_{k-2}, \dots, S_\alpha)$ and $s'' \cdot \sigma_{k-4}v' = s$. Since $S''_{k-1} = S_{k-1}$, $S''_{k-2} = S_{k-2}$ and $1 \in S''_{k-3}$, we are taken back to point (B3) (with $j_0 = 0$) and the desired couple (u, v) is obtained by composing (ν_k, σ_{k-2}) with (u', v') , which gives $(\nu_k u', \sigma_{k-2}\sigma_{k-4}v')$.
- If $1 \notin S_{k-1}$, $0 \in S_{k-2}$ and $1 \in S_{k-2}$, we distinguish two cases:
 - (IH3) If $k > 3$ then by induction hypothesis, there exists a couple $(u', v') \in (\Pi^*, (\Sigma_{k-4})^*)$ such that $s'' = u' \cdot s$ is a valid state of the form $(i'', S''_2, \dots, S''_{k-3}, S_{k-2} \cup \{1\}, S_{k-1}, \dots, S_\alpha)$ and $s'' \cdot \sigma_{k-3}v' = s$ (remark: if $k = 4$ then $v' = \varepsilon$). Since $S''_{k-1} = S_{k-1}$ and $0, 1 \in S''_{k-2}$, we are taken back to previous cases ((B3) or (IH2) according to S_{k-3} and S_{k-4}) which allow to find a couple (u'', v'') such that $u'' \cdot s''$ is a valid state and $(u'' \cdot s'') \cdot \sigma_{k-1}v'' = s''$. The desired couple (u, v) is obtained by composing $(u'', v'') \in (\Pi^*, (\Sigma_{k-2})^*)$ with (u', v') , which gives $(u''u', v''\sigma_{k-3}v')$.
 - (B4) If $k = 3$ then the desired couple (u, v) is (ν_3, ε) or (ν_3, σ_1) depending on whether $i > 0$ or $i = 0$. Indeed, $\nu_3 \cdot s$ is always a valid state (by lemma 6) of the announced form by definition of ν_3 and $(\nu_3 \cdot s) \cdot \sigma_2v = s$ by lemma 7 (applicable, because $1 \notin S_{k-1}$, i.e. $1 \notin S_2$).

Corollary 1. *For any valid state $s = (i, S_2, \dots, S_{\alpha-1}, \{0\})$ such that $q_f \notin S_{\alpha-1}$ there exists a valid state $s' = (i', S'_2, \dots, S'_{\alpha-1}, \{1\})$ and a word w such that $s' \cdot \sigma_{\alpha-1}w = s$.*

Proof. By previous lemma, there exists a state $s'' = (i'', S''_2, \dots, S''_{\alpha-1}, \{0, 1\})$ and a word $v \in (\Sigma_{\alpha-2})^*$ such that $s'' \cdot \sigma_{\alpha-1}v = s$. It is easy to see, because of the alphabet of v , that $q_f \in S''_{\alpha-1}$ if and only if $q_f \in S_{\alpha-1}$. Let us denote s''' the valid state such that $s'' \cdot \sigma_{\alpha-1} = s'''$. Since $q_f \notin S''_{\alpha-1}$, the sequence $(i'', S''_2, \dots, S''_{\alpha-1}, \{1\})$ is a valid state. Furthermore, we notice that this state verifies: $(i'', S''_2, \dots, S''_{\alpha-1}, \{1\}) \cdot \sigma_{\alpha-1} = s'''$. So, we deduce the desired state is $s' = (i'', S''_2, \dots, S''_{\alpha-1}, \{1\})$ and $w = v$.

Proposition 3. *Any valid state $s = (i, S_2, \dots, S_\alpha)$ is accessible.*

Proof. By induction on α , the base case being when $\alpha = 1$. In this case, the proposition is trivially verified since A_1 is a minimal DFA. Now, by induction hypothesis, we know that each valid state of the form $t = (i, S_2, \dots, S_{\alpha-1})$ is accessible. As we will see, if $q_f \in S_{\alpha-1}$, one can suppose verified the property (\mathcal{P}) stating that t is reached in the following way: $(0, \emptyset, \dots, \emptyset) \xrightarrow{\lambda_1} t_1 \xrightarrow{\lambda_2} t_2 \dots \xrightarrow{\sigma_\alpha} t_k \dots \xrightarrow{\lambda_\ell} t$ with $\forall t_{i(i < k)}, q_f \notin S_\alpha$ et $\forall j > k, \lambda_j \neq \sigma_\alpha$ (intuitively, when we reach the final state of $A_{\alpha-1}$ we no longer permute on this automaton). So, the induction hypothesis allows to suppose accessible, any valid state of the form $(i, S_2, \dots, S_{\alpha-1}, \emptyset)$, as well as any valid state of the form $(i, S_2, \dots, S_{\alpha-1}, \{0\})$ with $q_f \in S_{\alpha-1}$. To prove the accessibility of s , we follow a second induction based on the following partial order, defined on the possible sets $S_\alpha = \{k_0, k_1, \dots\}$:

$$S_\alpha < S'_\alpha \text{ if } |S_\alpha| < |S'_\alpha| \text{ or } \\ |S_\alpha| = |S'_\alpha| \text{ and } k_0 < k'_0 \text{ or } \\ |S_\alpha| = |S'_\alpha|, k_0 = k'_0 \text{ and } k_1 < k'_1.$$

We first prove that each state is accessible when S_α is a singleton. We set $S_{\alpha-1} = \{j_0, j_1, \dots\}$ and $S_\alpha = \{k_0\}$.

- If $j_0 > 0$ then $q_f \notin S_{\alpha-2}$. The state $s' = (i, S_2, \dots, S_{\alpha-2}, (\sigma_\alpha)^{j_0+1} \cdot S_{\alpha-1}, \{0\})$ is accessible by induction hypothesis (because $q_f \in (\sigma_\alpha)^{j_0+1} \cdot S_{\alpha-1}$). It is easy to verify that s' is valid (mainly because $q_f \notin S_{\alpha-2}$). If j_0 is odd then $s' \xrightarrow{(\sigma_\alpha)^{j_0+1}(\sigma_{\alpha+1})^{k_0}} s$. If j_0 is even then $s' \xrightarrow{(\sigma_\alpha)^2\sigma_{\alpha+1}(\sigma_\alpha)^{j_0-1}(\sigma_{\alpha+1})^{k_0}} s$.
- If $j_0 = 0$ and $k_0 > 0$, we distinguish two cases:
 - If $q_f \notin S_{\alpha-2}$ or $j_1 = 1$ then the state $s' = (i, S_2, \dots, S_{\alpha-2}, \sigma_\alpha \cdot S_{\alpha-1}, \{0\})$ is valid and accessible by induction hypothesis (because $q_f \in \sigma_\alpha \cdot S_{\alpha-1}$) and $s' \xrightarrow{\sigma_\alpha(\sigma_{\alpha+1})^{k_0-1}} s$.
 - If $q_f \in S_{\alpha-2}$ (and so $0 \in S_{\alpha-1}$) and $1 \notin S_{\alpha-1}$ then by Lemma 8, there exists a valid state $s' = (i', S'_2, \dots, S'_{\alpha-2}, S_{\alpha-1} \cup \{1\}, \{k_0\})$ and a word v such that $s' \xrightarrow{v} s$. And s' is accessible following the previous point.
- If $j_0 = 0$ and $k_0 = 0$ then, either $q_f \in S_{\alpha-1}$ and s is accessible by induction hypothesis, or $q_f \notin S_{\alpha-1}$ and, by Corollary 1, there exists a valid state $s' = (i', S'_2, \dots, S'_{\alpha-1}, \{1\})$ and a word w such that $s' \xrightarrow{w} s$. And s' is accessible following one of the previous points.

Now, we look at the case where S_α contains at least two states.

- If $k_0 > 0$ then the state $s' = (i, S_2, \dots, S_{\alpha-1}, \sigma_{\alpha+1} \cdot S_\alpha)$ is valid and accessible by induction hypothesis (we have decreased by 1 the first index of S_α) and $s' \cdot \sigma_{\alpha+1} = s$.
- If $k_0 = 0$ and $k_1 = 1$, we distinguish two cases:
 - If $q_f \notin S_{\alpha-2}$ or $1 \in S_{\alpha-1}$ then the state $s' = (i, S_2, \dots, S_{\alpha-2}, (\sigma_\alpha)^{j_0+1} \cdot S_{\alpha-1}, \sigma_{\alpha+1} \cdot (S_\alpha \setminus \{0\}))$ is valid and accessible by induction hypothesis (the last set contains one less state than S_α) and $s' \xrightarrow{\sigma_{\alpha+1}(\sigma_\alpha)^{j_0+1}} s$.
 - If $q_f \in S_{\alpha-2}$ (which implies $0 \in S_{\alpha-1}$) and $1 \notin S_{\alpha-1}$ then, by Lemma 8, there exists a valid state $s' = (i', S'_2, \dots, S'_{\alpha-2}, S_{\alpha-1} \cup \{1\}, S_\alpha)$ and a word w such that $s' \xrightarrow{w} s$. And s' is accessible following the previous point.
- If $k_0 = 0$ and $k_1 > 1$ then, by Lemma 8, there exists a valid state $s' = (i', S'_2, \dots, S'_{\alpha-1}, S_\alpha \cup \{1\})$ and a word w such that $s' \xrightarrow{w} s$. And s' is accessible following the previous case.

One can verify that, in each of the considered cases, we never act in a final valid state (*i.e.* (S_1, \dots, S_α) with $q_f \in S_\alpha$) with a $\sigma_{\alpha+1}$ letter. This ensures the property \mathcal{P} announced at the beginning of the proof.

From Propositions 3 and 2, we deduce:

Theorem 2. *The family of sequences of minimal DFAs $(A_1, \dots, A_\alpha)_{\alpha \geq 0}$, described in table 1, is a family of witnesses over an $(\alpha + 1)$ -letters alphabet for the catenation of α languages.*

5 A α -letters witness for the catenation of α automata: a conjecture

In this section we propose to decrease by one the size of the alphabet used to define witnesses for multiple catenation. A α -letters alphabet should be optimal. In any case, it is optimal when $\alpha = 2$: indeed it is proven in [12] that state complexity for catenation of two minimal DFAs with size m and n , and using only one letter is mn , which is strictly lower to the general state complexity for catenation.

Our statement is a conjecture, since we only prove it when $\alpha = 2$ and $\alpha = 3$. Some tests computed with the software Sage for $\alpha \in [2, 7]$ and DFAs with size in [3, 6] also argue in this sense. Our witnesses can be obtained by slightly modifying the table of the previous section (see Table 2).

5.1 The two automata case

In its paper [1], where J. Brzozowski proposes four atomic constructions to build universal witnesses, he observes a defect concerning the operation of catenation. He only suggests a 3-letters alphabet witness, whereas, in [10], G. Jiraskova produces a 2-letters one.

We give here a 2-letters witness for catenation, based on the atomic constructions of J. Brzozowski, which corresponds to the previous table when $\alpha = 2$ (see Table 3 and Figure 3).

	A_1	A_2	A_3	\dots	A_α
σ_1	t	c	$\mathbb{1}$		$\mathbb{1}$
σ_2	p	t	c		$\mathbb{1}$
σ_3	$\mathbb{1}$	p	t	c	
\vdots					
σ_α	$\mathbb{1}$				p

Table 2. α -letters witnesses for the multiple catenation: a conjecture

	A_1	A_2
b	t	c
a	p	p

Table 3. 2-letters witness for catenation of two languages

Following Definition 2, we add transitions from the predecessor of the final state in the first DFA to the initial state of the second DFA and apply the subset construction to the resulting NFA. The valid states of this automaton, named A in the following, are of the form (p_i, S) , where S denotes any subset of $\{q_0, \dots, q_{n-1}\}$ (containing q_0 if $i = m - 1$). The number of valid states is equal to the state complexity of catenation, that is $m2^n - 2^{n-1}$. We prove that all these states are both accessible and pairwise non-equivalent.

Proposition 4. *Each valid state (p_i, S) of A is accessible.*

Proof. By induction on the size of S . First, any state of the form (p_i, \emptyset) with $i \neq m - 1$ is accessible from the initial state (p_0, \emptyset) by the word a^i .

Now, consider any state $s = (p_i, S)$ with $|S| = k$ for some integer $k > 0$. We proceed by cases :

1. If $i = m - 1$ then $q_0 \in S$ and s is reached by a from $(p_{m-2}, a.(S \setminus \{q_0\}))$, which is accessible by induction hypothesis.
2. If $i = 0$ and $q_1 \in S$, s is reached by a from $(m - 1, a.S)$ which is accessible by point 1.
3. If $i = 0$ and $q_1 \notin S$. Let us set $q_1 < \dots < q_{n-1} < q_n = q_0$ and S be an ordered set with $S = \{q_{j_1}, \dots, q_{j_\alpha}\}$. Then s is reached by $(ab)^{j_1-1}$ from $(0, \{q_1, \dots, q_{j_\alpha-j_1+1}\})$ which is accessible by point 2.
4. If $i \in]0, m - 1[$ then s is reached by a^i from $(0, a^i.S)$ which is accessible by one of the previous two points.

Proposition 5. *Any two distinct valid states $s = (p_i, S)$ and $s' = (p_{i'}, S')$ of A are non-equivalent.*

Proof. There are two cases to consider:

- If $S \neq S'$, without loss of generality, let $q_j \in S \setminus S'$. Then a^{n-1-j} sends s to a final state and s' to a non-final one.
- Now, suppose $S = S'$. So $i \neq i'$. By reading the word $a^{(m-n-i) \bmod m}$, one sends s to (p_{i_1}, S_1) with $i_1 = (m - n) \bmod m$, and s' to $(p_{i'_1}, S'_1)$ with $i'_1 \neq i_1$. Then, by the word bb , we send (p_{i_1}, S_1) to $(p_{i_1}, S_1 \setminus \{q_1\})$ and $(p_{i'_1}, S'_1)$ to $(p_{i'_1}, S'_1 \setminus \{q_1\})$. Last, by reading the word a^{n-1} , we send $(p_{i_1}, S_1 \setminus \{q_1\})$ to (p_{m-1}, S_2) and $(p_{i'_1}, S'_1 \setminus \{q_1\})$ to $(p_{i'_2}, S'_2)$ with $q_0 \in S_2 \setminus S'_2$. So we have reduced this case to the previous one.

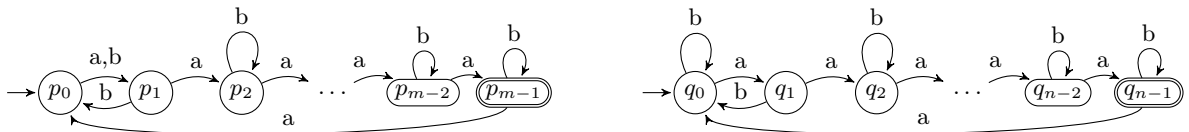


Fig. 3. The witness described by Table 3.

It follows from Propositions 4 and 5 that:

Theorem 3. *The couple of Brzozowski automata defined in Table 3 is a 2-letters witness for the catenation of two languages.*

5.2 The three automata case

The triple of Brzozowski automata A_1, A_2, A_3 with respective size m, n, p described in Figure 4 is the proposed 3-letters witness for the double catenation.

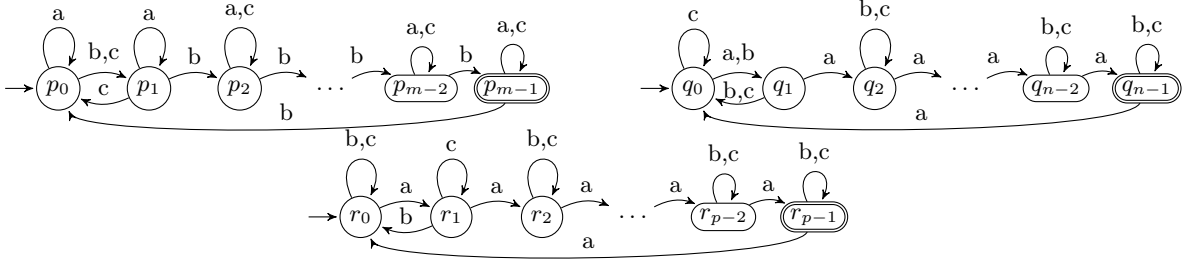


Fig. 4. 3-letters witness for double catenation

It corresponds to Table 2 when $\alpha = 3$ (see Table 4).

	A_1	A_2	A_3
c	t	c	$\mathbf{1}$
b	p	t	c
a	$\mathbf{1}$	p	p

Table 4. 3-letters witness for catenation of three languages

The accessible states of A (the DFA obtained by the subset algorithm from A_1, A_2 and A_3 connected as described in Definition 2) are identified to 3-tuples of the form $(p_i, S = \{q_{j_0}, \dots, q_{j_\beta}\}, T = \{r_{k_0}, \dots, r_{k_\gamma}\})$, with $j_0 < \dots < j_\beta$ and $k_0 < \dots < k_\gamma$, and must satisfy the three following constraints:

- $i = m - 1 \Rightarrow q_0 \in S$.
- $q_{n-1} \in S \Rightarrow r_0 \in T$.
- $S = \emptyset \Rightarrow T = \emptyset$.

These constraints corresponds to the properties $P1, P2$ and $P3$ in the peculiar case where $\alpha = 3$. So the number of states verifying these constraints (valid states) is equal to $\#\mathcal{T}_3$, the value computed in Example 1. We prove all these states are both accessible and pairwise non-equivalent.

Proposition 6. *Any valid state $s = (p_i, S, T)$ of A is accessible.*

Proof. By induction over the size of $S \cup T$. First, any state of the form $(p_i, \emptyset, \emptyset)$ is accessible by b^i from the initial state $(p_0, \emptyset, \emptyset)$. Next, consider some integer θ and suppose any state (p_i, S, T) with $|S \cup T| \leq \theta$ is accessible. We prove, by cases, that any state (p_i, S, T) with $|S \cup T| = \theta + 1$ is also accessible.

1. $i = m - 1$
 - (a) $S = \{q_0\}$ and $T = \emptyset$
 $(p_{m-2}, \emptyset, \emptyset) \xrightarrow{b} s$

(b) $S = \{q_0\}$ and $T \neq \emptyset$

$$\begin{aligned} (p_{m-1}, \{q_0\}, a^{n-1+k_0} \cdot (T \setminus \{r_{k_0}\})) &\xrightarrow{a} \\ (p_{m-1}, \{q_0, q_1\}, a^{n-2+k_0} \cdot (T \setminus \{r_{k_0}\})) &\xrightarrow{(ac)^{n-2}} \\ (p_{m-1}, \{q_0, q_{n-1}\}, a^{k_0} \cdot T) &\xrightarrow{(ac)^{k_0+p}} s \end{aligned}$$

The suffix $(ac)^{k_0}$ is sufficient in general, except when $k_0 = 0$, since one last occurrence of ac is necessary over A_2 .

(c) $|S| > 1$

If $j_1 > 1$ then

$$\begin{aligned} (p_{m-1}, a^{j_1} \cdot (S \setminus \{q_0\}), a^{j_1} \cdot T) &\xrightarrow{a} \\ (p_{m-1}, \{q_0\} \cup a^{j_1-1} \cdot (S \setminus \{q_0\}), a^{j_1-1} \cdot T) &\xrightarrow{ac} \\ (p_{m-1}, \{q_0\} \cup a^{j_1-2} \cdot (S \setminus \{q_0\}), a^{j_1-2} \cdot T) &\xrightarrow{(ac)^{j_1-2}} s \end{aligned}$$

if $j_1 = 1$ then

$$(p_{m-1}, a \cdot (S \setminus \{q_0\}), a \cdot T) \xrightarrow{a} s$$

2. $i < m-1$

(a) $S = \{q_{j_0}\}$ and $T = \emptyset$

$$\begin{aligned} (p_{m-2}, \emptyset, \emptyset) &\xrightarrow{b^{i+2}} \\ (p_i, \{q_{(i+1) \bmod 2}\}, \emptyset) &\xrightarrow{c^2} \\ (p_i, \{q_{j_0}\}, \emptyset) &\xrightarrow{a^{j_0}} s \end{aligned}$$

(b) $S = \{q_{j_0}\}$ and $T \neq \emptyset$

Let $\delta = (k_0 - j_0) \bmod p$ and $i' = \begin{cases} i & \text{if } i > 1 \text{ or } \delta + p \text{ is even} \\ 1 - i & \text{otherwise} \end{cases}$

$$\begin{aligned} (p_{i'}, \{q_{n-2}\}, a^{k_0+1} \cdot (T \setminus \{r_{k_0}\})) &\xrightarrow{a} \\ (p_{i'}, \{q_{n-1}\}, \{r_0\} \cup a^{k_0} \cdot (T \setminus \{r_{k_0}\})) &\xrightarrow{(ac)^{\delta+p}} \\ (p_i, \{q_0\}, \{r_\delta\} \cup a^{(-j_0) \bmod p} \cdot (T \setminus \{r_{k_0}\})) &\xrightarrow{a^{j_0}} s \end{aligned}$$

As previously, the factor $(ac)^p$ ensures one occurrence of ac even when $(k_0 - j_0) \bmod p = 0$.

(c) $|S| > 1$ and $r_{j_0+1} \notin T$

i. $j_1 > j_0 + 1$

$$\begin{aligned} (p_{m-2}, a^{j_0} \cdot (S \setminus \{q_{j_0}\}), a^{j_0} \cdot T) &\xrightarrow{bb} \\ (p_0, \{q_1\} \cup a^{j_0} \cdot (S \setminus \{q_{j_0}\}), a^{j_0} \cdot T) &\xrightarrow{b^i} \\ (p_i, \{q_{(1+i) \bmod 2}\} \cup a^{j_0} \cdot (S \setminus \{q_{j_0}\}), a^{j_0} \cdot T) &\xrightarrow{cc} \\ (p_i, \{q_0\} \cup a^{j_0} \cdot (S \setminus \{q_{j_0}\}), a^{j_0} \cdot T) &\xrightarrow{a^{j_0}} s \end{aligned}$$

ii. $j_1 = j_0 + 1$ (we have $q_0, q_1 \in a^{j_0} \cdot S$)

$$\begin{aligned} (p_{m-2}, a^{j_0} \cdot (S \setminus \{q_{j_1}\}), a^{j_0} \cdot T) &\xrightarrow{bb} \\ (p_{m-2}, a^{j_0} \cdot S, a^{j_0} \cdot T) &\xrightarrow{b^i} \\ (p_i, a^{j_0} \cdot S, a^{j_0} \cdot T) &\xrightarrow{a^{j_0}} s \end{aligned}$$

(d) $|S| > 1$ and $r_{j_0+1} \in T$

If $S = Q_{A_2}$ then note that $r_0, r_1 \in T$ and so: $(p_i, S, a \cdot (T \setminus \{r_0\})) \xrightarrow{a} s$.

If $S \neq Q_{A_2}$ then we set $\Delta = \min\{\delta > 0 \mid q_{j_0+\delta} \notin S\}$. If $\Delta = 1$ then we first notice that $q_{n-1} \notin a^{j_0+2} \cdot S$ and $q_{n-1} \in a^{j_0+1} \cdot S$. Hence, $(p_i, a^{j_0+2} \cdot S, a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\}))$ is valid and

$$(p_i, a^{j_0+2} \cdot S, a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{a} (p_i, a^{j_0+1} \cdot S, a^{j_0+1} \cdot T)$$

because $r_{p-1} \notin a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\})$. Furthermore, since $j_0 = \min\{j \mid q_j \in S\}$, we have

$$(p_i, a^{j_0+1} \cdot S, a^{j_0+1} \cdot T) \xrightarrow{a^{j_0+1}} s.$$

Now let us examine the case when $\Delta > 1$ and set $R = \{r_{j_0+2}, \dots, r_{j_0+\Delta}\}$. We consider two situations:

i. $R \cap T = \emptyset$.

- A. $i \neq (\Delta - 2) \bmod m$
 Since $q_{n-1} \in a^{j_0+\Delta} \cdot S$ we have

$$(p_{i-(\Delta-1)}, a^{j_0+\Delta+1} \cdot S, a^{j_0+\Delta+1} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{a} (p_{i-(\Delta-1)}, a^{j_0+\Delta} \cdot S, \{r_0\} \cup a^{j_0+\Delta} \cdot (T \setminus \{r_{j_0+1}\})).$$

But $q_0 \in a^{j_0+\Delta-1} \cdot S$ so

$$(p_{i-(\Delta-1)}, a^{j_0+\Delta} \cdot S, \{r_0\} \cup a^{j_0+\Delta} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{abcc} (p_{i-(\Delta-2)}, a^{j_0+\Delta-1} \cdot S, \{r_0\} \cup a^{j_0+\Delta-1} \cdot (T \setminus \{r_{j_0+1}\})).$$

Furthermore

$$(p_{i-(\Delta-2)}, a^{j_0+\Delta-1} \cdot S, \{r_0\} \cup a^{j_0+\Delta-1} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{(ab)^{\Delta-3}} (p_{i-1}, a^{j_0+2} \cdot S, \{r_0\} \cup a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\})).$$

But $r_{p-1} \notin a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\})$ implies

$$(p_{i-1}, a^{j_0+2} \cdot S, \{r_0\} \cup a^{j_0+2} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{ab} (p_i, a^{j_0+1} \cdot S, a^{j_0+1} \cdot T).$$

And finally,

$$(p_i, a^{j_0+1} \cdot S, a^{j_0+1} \cdot T) \xrightarrow{a^{j_0+1}} s.$$

- B. $i = (\Delta - 2) \bmod m$

We proceed in a very similar way excepting that we start on p_{m-2} on the first automaton rather than $p_{i-(\Delta-1)} = p_{m-1}$ and we use a slightly different prefix $aabcbcc$ instead of $aabcc$:

$$(p_{m-2}, a^{j_0+\Delta+1} \cdot S, a^{j_0+\Delta+1} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{aabcbcc(ab)^{\Delta-2}a^{j_0+1}} s.$$

- ii. $R \cap T \neq \emptyset$.

We set $\Psi = \min\{\psi > 0 \mid r_{j_0+1+\psi} \in T\}$. The proof goes as in the previous case (i) but we do not need to contract with c on A_2 . We construct the good words by deleting the letters c and replacing Δ by Ψ . So the last two cases are

- A. $i \neq (\Psi - 2) \bmod m$

$$(p_{i-(\Psi-1)}, a^{j_0+\Psi+1} \cdot S, a^{j_0+\Psi+1} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{aab(ab)^{\Psi-2}a^{j_0+1}} s$$

- B. $i = (\Psi - 2) \bmod m$

$$(p_{m-2}, a^{j_0+\Psi+1} \cdot S, a^{j_0+\Psi+1} \cdot (T \setminus \{r_{j_0+1}\})) \xrightarrow{aabb(ab)^{\Psi-2}a^{j_0+1}} s$$

Proposition 7. Any two distinct valid states, $s = (p_i, S, T)$ and $s' = (p_{i'}, S', T')$ of A are non-equivalent.

Proof. There are three cases to consider:

- If $T \neq T'$, without loss of generality, let $r_k \in T \setminus T'$. Then a^{p-1-k} sends s to a final state and s' to a non-final one.
- Now, suppose $T = T'$ and $S \neq S'$. Without loss of generality, let $q_j \in S \setminus S'$. First, read the word a^{n-1-j} . It sends s to $s_1 = (p_i, S_1, T_1)$ and s' to $s'_1 = (p_{i'}, S'_1, T'_1)$ with $q_{n-1} \in S_1 \setminus S'_1$. Then, read the word $b^{m-i'}$ to send s_1 to $s_2 = (p_{i_2}, S_2, T_2)$ and s'_1 to $s'_2 = (p_0, S'_2, T'_2)$ with $q_{n-1} \in S_2 \setminus S'_2$. Now, we set

$$t = \begin{cases} (n-p+1) \bmod n & \text{if } (n-p) \bmod n \neq 0, 1 \\ 2 - (n-p) & \text{otherwise} \end{cases}$$

By reading the word $a^t b$, one sends s_2 to $s_3 = (p_{i_3}, S_3, T_3)$ and s'_2 to $s'_3 = (p_1, S'_3, T'_3)$ with $q_{n-p} \in S_3 \setminus S'_3$ and $r_1 \notin T_3 \cup T'_3$. Last, by reading the word a^{p-1} , we send s_3 to (p_{i_3}, S_4, T_4) and s'_3 to (p_1, S'_4, T'_4) with $r_0 \in T_4 \setminus T'_4$. So we have reduced this case to the previous one.

- Last, if $T = T'$, $S = S'$ and $i \neq i'$, without loss of generality, let us suppose that $i > i'$. Then by reading the word $b^{m-1-i}ca^{n-1}$, we send s to (p_{m-1}, S_1, T_1) and s' to $(p_{i'}, S'_1, T'_1)$ with $q_0 \in S_1 \setminus S'_1$. That is we have reduced this case to one of the two previous ones.

It follows from Propositions 6 and 7 that:

Theorem 4. The triple of Brzozowski automata A_1, A_2, A_3 defined in Table 4 is a 3-letters witness for the catenation of three languages.

6 Conclusion

In this paper we have dramatically reduced the size of the alphabet needed to produce a family of witnesses for multiple catenation: $(\alpha + 1)$ -letters alphabet witness for catenation of α languages. We obtain this result by using Brzozowski DFAs, giving some new evidence of the fact that these tools seems a very good starting point to discover witnesses. We also give a simple recursive formulae for the bound. Its effective computation gives rise to a combinatorial expression involving compositions which is an efficient alternative to the formulae given by Gao and Yu [8] in the optimal case where automata have only one final state.

It remains, at least, two open problems:

1. The proof of the conjecture given in the last section where a α -letters alphabet witnesses is given for catenation of α languages, but only validated for $\alpha = 2, 3$.
2. The optimality of the size of the alphabet. Clearly, it is true when $\alpha = 2$ but is it still true for greater values ?

References

1. Janusz A. Brzozowski. In search of most complex regular languages. *Int. J. Found. Comput. Sci.*, 24(6):691–708, 2013.
2. Pascal Caron, Jean-Gabriel Luque, Ludovic Mignot, and Bruno Patrou. State complexity of catenation combined with a boolean operation: a unified approach. <http://arxiv.org/abs/1505.03474>, abs/1505.03474, 2015. To appear in IJFCS.
3. Bo Cui, Yuan Gao, Lila Kari, and Sheng Yu. State complexity of two combined operations: Catenation-union and catenation-intersection. *Int. J. Found. Comput. Sci.*, 22(8):1797–1812, 2011.
4. O Ganyushkin and Volodymyr Mazorchuk. *Classical finite transformation semigroups: an introduction*. Algebra and Applications. Springer, Dordrecht, 2008.
5. Yuan Gao, Nelma Moreira, Rogério Reis, and Sheng Yu. A survey on operational state complexity. *CoRR*, abs/1509.03254, 2015.
6. Yuan Gao, Nelma Moreira, Rogério Reis, and Sheng Yu. A review on state complexity of individual operations. technical report DCC-2011-08, University of Porto, 2012.
7. Yuan Gao, Kai Salomaa, and Sheng Yu. The state complexity of two combined operations: Star of catenation and star of reversal. *Fundam. Inform.*, 83(1-2):75–89, 2008.
8. Yuan Gao and Sheng Yu. State complexity approximation. In Jürgen Dassow, Giovanni Pighizzini, and Bianca Truthe, editors, *Proceedings Eleventh International Workshop on Descriptive Complexity of Formal Systems, DCFS 2009, Magdeburg, Germany, July 6-9, 2009.*, volume 3 of *EPTCS*, pages 121–130, 2009.
9. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.
10. Galina Jirásková. State complexity of some operations on binary regular languages. *Theor. Comput. Sci.*, 330(2):287–298, 2005.
11. Galina Jirásková and Alexander Okhotin. On the state complexity of star of union and star of intersection. *Fundam. Inform.*, 109(2):161–178, 2011.
12. Sheng Yu, Qingyu Zhuang, and Kai Salomaa. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125(2):315–328, 1994.